

Pseudo derivative evolutionary algorithm and convergence analysis

Yu, Yang; Wang, Zhongjie; Tianfield, Huaglory; Lu, Chengchao

Published in:
International Journal of Modeling, Simulation, and Scientific Computing

DOI:
[10.1142/S1793962318500411](https://doi.org/10.1142/S1793962318500411)

Publication date:
2018

Document Version
Author accepted manuscript

[Link to publication in ResearchOnline](#)

Citation for published version (Harvard):
Yu, Y, Wang, Z, Tianfield, H & Lu, C 2018, 'Pseudo derivative evolutionary algorithm and convergence analysis', *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 9, no. 5.
<https://doi.org/10.1142/S1793962318500411>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please view our takedown policy at <https://edshare.gcu.ac.uk/id/eprint/5179> for details of how to contact us.

Pseudo Derivative Evolutionary Algorithm and Convergence Analysis

Yang Yu^{a,1}, Zhongjie Wang^{b,2,*}, Huaglory Tianfield^{c,3}

^aDepartment of Control Science and Engineering, Tongji University, Shanghai 201804, China

^bDepartment of Control Science and Engineering, Tongji University, Shanghai 201804, China

^cDepartment of Computer, Communications and Interactive Systems, Glasgow Caledonian University, Glasgow G4 0BA, United Kingdom

Abstract

In this paper a novel evolutionary algorithm (EA), called pseudo derivative evolutionary algorithm (PDEA), is proposed. The basic idea of PDEA is to use pseudo derivative, which is obtained based on the information produced during the evolution, to help search the solution of optimization problem. The pseudo derivative drives the search process in a more informed direction. That makes PDEA different from the random optimization methods. The convergence of PDEA is first analyzed based on systems theory. The convergence condition of PDEA is then derived, though this condition is too strong to be satisfied. Next, this condition is relaxed based on entropy theory. Finally, performances of PDEA are evaluated on the benchmark functions and an adaptive liquid level control system of a surge tank. The numeric simulation results show that PDEA is capable of finding the solutions to the optimization problems with good accuracy, reliability, and speed.

Keywords: pseudo derivative evolutionary algorithm (PDEA), evolutionary algorithm (EA), convergence analysis, entropy theory

1. Introduction

Evolutionary algorithm (EA) is a kind of population algorithm to search for an optimal or near optimal solution to complex problems in polynomial time. Common variants of EA include genetic algorithm (GA) [1–3], genetic programming [4] [5], differential evolution (DE) [6] [7], particle swarm optimization (PSO) [8–10], bacterial foraging optimization (BFO) [11–13], artificial bee colony algorithm (ABC) [14] etc.

EAs do not require gradient information. Therefore, they are capable of solving a wide variety of optimization problems, which may be linear, quadratic, unimodal, discontinuous, non-differentiable, strongly convex, etc [15–18]. However, different from derivative based iterative methods for optimization, the convergence of EAs have not proved explicitly, or is proved under some very strong hypothesis.

To analyse the convergence of GA, a Markov chain model is proposed [19–22]. The process of GA can be proved as a homogeneous Markov chain. Then GA is analysed based on the properties and theorems of the Markov chain. GA is proved to be convergent with the assumption that the number of generations is infinite. PSO can be modelled by dynamical equations [23–26]. The evolution process of PSO can be divided into two

different parts. The first part assumes each individual has an initial position and velocity. The second part is about how to produce the next generation individuals. The velocity is modified by two kinds of information, i.e., the best position in all preceding generations and the best position the individual ever obtained. With iterations of these two parts, PSO is able to find the solution to an optimization problem. By modelling the iteration process, the dynamic equations of PSO can be formulated. Thereafter, PSO is analysed based on the dynamic equations. However, the convergence condition of PSO is too strong that even the standard PSO is not convergent. BFO, modelling the individual and group behaviour of *E.Coli* bacteria, is a distributed optimization process. Reference [27] analyses BFO by formulating the mathematical model of the chemotactic movements in continuous time. Then the Lyapunov stability theorem is used to analyse the convergence of the dynamic model. It assumes the objective function is continuously differentiable. References [28] [29] analyse the one-dimensional DE based on a mathematical model. The model has been formulated based on probability theory and dynamic equation. It assumes the trial solutions are limited within a small region, and the fitness landscape has a moderate gradient.

This paper proposes a pseudo derivative EA (PDEA), which explicitly uses the pseudo derivative obtained based on the information produced during the evolution process to search for the optimal solution. First of all, the convergence of PDEA is studied based on systems theory. PDEA is proved to be convergent. The convergence condition is derived, though this condition is too strict and hardly satisfied. Then, we relax the convergence condition based on entropy theory.

The remainder of this paper is arranged as follows. The principle of PDEA is presented in Section 2. In Section 3, con-

*Zhongjie Wang

Email addresses: yuyang_box@qq.com (Yang Yu),
wang_zhongjie@tongji.edu.cn (Zhongjie Wang),
H.Tianfield@gcu.ac.uk (Huaglory Tianfield)

¹Yang Yu is with the Department of Control Science and Engineering, Tongji University, Shanghai, China.

²Zhongjie Wang is a Professor with the Department of Control Science and Engineering, Tongji University, Shanghai, China.

³Huaglory Tianfield is a Professor of Computing with Glasgow Caledonian University, Glasgow, United Kingdom

vergence analysis is conducted to PDEA based on systems theory and entropy theory. In Section 4, performance evaluations are conducted by applying PDEA to benchmark functions and an adaptive liquid level control system. Finally, conclusions are drawn in Section 5.

2. PDEA

2.1. Pseudo Derivative

Iterative methods for optimization, such as Newton's method, gradient descent method, employ derivative of the objective function in searching the solution. By analogy, we introduce the concept of pseudo derivative to EAs. The basic idea of PDEA is to make use of a pseudo derivative information explicitly in searching the solution to the optimization problem.

The information produced during the evolution contains the individual position and fitness value in preceding and current generations, and the distance between two individuals. Therefore, we define a pseudo derivative as below.

$$pdx = \frac{f(i) - f(j)}{\|pos(i) - pos(j)\|} \quad i \neq j \quad (1)$$

where i, j are indices of individuals, $f(i)$ and $pos(i)$ denote the fitness value and position of individual i within the search space. $\|pos(i) - pos(j)\|$ is the distance between individual i and individual j . The right-hand side of Eq.1 means the average change of fitness value over $\|pos(i) - pos(j)\|$. Actually, we can find that $\lim_{\|pos(i) - pos(j)\| \rightarrow 0} pdx$ is the derivative in the direction of vector $(pos(i) - pos(j))$. Therefore, pdx is called the pseudo derivative.

Using the best fitness values in the current generation and across all the preceding generations, we have

$$pdc = \frac{f_{cbest} - f(i)}{\|p_{cbest} - pos(i)\|} \quad (2)$$

$$pdh = \frac{f_{hbest} - f(i)}{\|phbest - pos(i)\|} \quad (3)$$

where f_{cbest} denotes the best fitness value in current generation, p_{cbest} the corresponding position. f_{hbest} and $phbest$ denote the best fitness value and position across all preceding generations. pdc and pdh are the pseudo derivatives which represent the average fitness change in the direction of an individual towards the best individual. Compared with true derivative, pdc and pdh represent the approximations of fitness value changes during the search process in the directions towards p_{cbest} and $phbest$, respectively.

2.2. Algorithm

Now, we have two kinds of pseudo derivative informations pdc and pdh in PDEA. Different from true derivative, pseudo derivatives just reflect the approximation of the fitness value changes in two certain directions, but still, pdc and pdh can provide the information of objective function. In order to expand the search region and avoid premature convergence, we encourage the individual to explore its search region. For this

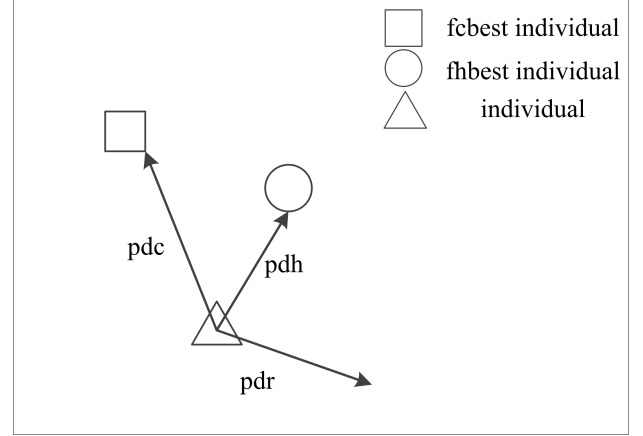


Figure 1: Pseudo derivatives of individual

purpose, we design the algorithm such that each individual has a random direction which is defined by pdr with a step to move within the neighbourhood, as illustrated in Fig.1.

The individual behaviour is driven by the three kinds of pseudo derivatives, i.e., pdc , pdh , and pdr . Regardless of the driving strategies, the procedure of PDEA can be described as in Algorithm 1.

Algorithm 1 PDEA

Input: maximum number of iterations N_{max} , and $k = 0$

- 1: generate the initial individuals and calculate the fitness values;
- 2: **for** $k \leq N_{max}$ **do**
- 3: obtain pdc , pdh , and pdr of every individual;
- 4: update individuals according to the driving strategy and form the new generation
 $new\ individual := S_k(individual, pdc, pdh, pdr)$;
- 5: **end for**

Output: the best individual and its fitness value;

Different driving strategies defined by $S(\cdot)$ will result in different paradigms of PDEA. The driving strategies $S(\cdot)$ used to update individuals can take various forms. Without loss of generality, here we consider the strategy as follows.

$$S_{k+1}(i) = \alpha \cdot u_1 \cdot (p_{cbest}(k) - pos_k(i)) + \beta \cdot u_2 \cdot (phbest(k) - pos_k(i)) + \gamma \cdot u_3 \cdot S_k(i) \quad (4)$$

$$S_1(i) = \mathbf{d} \quad (5)$$

$$pos_{k+1}(i) = pos_k(i) + S_{k+1}(i) \quad (6)$$

where α , β , and γ are the weight coefficients in the direction of pdc , pdh , and S_k . S_1 is determined by pdr . The value of α , β , and γ are determined by pdc , pdh , and pdr . \mathbf{d} is a random vector for the local search strategy with a given length. k is the index of iterations. $pos_k(i)$ denotes the position of individual i in generation k . $p_{cbest}(k)$, $phbest(k)$ denote the position of the best individual in current generation k and all preceding

generations prior to k . $\mathbf{u} = (u_1, u_2, u_3)$ represents the probability weight coefficients. Each element of \mathbf{u} can be a probability distribution or a logistic variable for determining whether to use the corresponding pseudo derivative. For example, if $\mathbf{u} = (0, 0, 1)$, $S(i, pdc, pdh, pdr)$ it is the same as the mutation part of DE/rand/1 algorithm [30].

3. Convergence Analysis of PDEA

3.1. Proof of PDEA convergence

Considering Eqs. (4) (5) (6), for individual i in generation k , let's use notations omitting the individual's index, denote $x(k) = S_k(i)$, $y(k) = pos_k(i)$. Then we have

$$x(k+1) = \varphi_1(pcbest(k) - y(k)) + \varphi_2(phbest(k) - y(k)) + \varphi_3x(k) \quad (7)$$

$$y(k+1) = y(k) + x(k+1) \quad (8)$$

$$\varphi_1 = \alpha \cdot u_1 \quad (9)$$

$$\varphi_2 = \beta \cdot u_2 \quad (10)$$

$$\varphi_3 = \gamma \cdot u_3 \quad (11)$$

By arranging Eqs. (7) and (8) into matrix form, we will have the state space model of PDEA as below.

$$\begin{bmatrix} x(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} \varphi_3 & -\zeta \\ \varphi_3 & 1-\zeta \end{bmatrix} \begin{bmatrix} x(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} \zeta \\ \zeta \end{bmatrix} \Gamma \quad (12)$$

where

$$\zeta = \varphi_1 + \varphi_2 \quad (13)$$

$$\Gamma = \frac{\varphi_1 \cdot pcbest(k) + \varphi_2 \cdot phbest(k)}{\varphi_1 + \varphi_2} \quad (14)$$

Eq. (12) describes PDEA as a dynamic system, $[x(k), y(k)]^T$ is its augmented state vector, where T denotes transposition and Γ is the control input.

Now, we can analyse the convergence of PDEA by system of Eq. (12). From the point of view of systems theory, if system of Eq. (12) is controllable, no matter what the desired state is, there must be a control that drives the system to there. That means the system can reach any point one wishes by means of a suitable input. In this case, we say the system is convergent. Let's denote

$$\mathbf{z}(k) = [x(k), y(k)]^T \quad (15)$$

$$\mathbf{A} = \begin{bmatrix} \varphi_3 & -\zeta \\ \varphi_3 & 1-\zeta \end{bmatrix} \quad (16)$$

$$\mathbf{B} = [\zeta, \zeta]^T \quad (17)$$

Definition 3.1. The discrete-time state space system of Eq. (12) is said controllable if for any initial state $\mathbf{z}(0) = \mathbf{z}_0$ and any target state \mathbf{z}_t , there exists an input series that transfers \mathbf{z}_0 to \mathbf{z}_t . Otherwise, the system is said uncontrollable.

Lema 3.1. The discrete-time state space system of Eq. 12 is controllable, if the controllability matrix

$$\mathbf{C} = [\mathbf{B} \ \mathbf{A}\mathbf{B} \ \mathbf{A}^2\mathbf{B} \ \dots \ \mathbf{A}^{n-1}\mathbf{B}] \quad (18)$$

has a full row rank of n , where n is the dimension of matrix \mathbf{A} .

Therefore, we have

$$\mathbf{C}_z = \begin{bmatrix} \zeta & \varphi_3\zeta - \zeta^2 \\ \zeta & \varphi_3\zeta - \zeta^2 + \zeta \end{bmatrix} \quad (19)$$

It is clear that matrix \mathbf{C}_z is of full rank when $\zeta \neq 0$. So, system of Eq. (12) is controllable, that is, it is convergent.

3.2. Convergence condition of PDEA

Consider the autonomous system

$$\mathbf{z}(k+1) = \mathbf{A}\mathbf{z}(k) \quad (20)$$

Definition 3.2. System $\mathbf{z}(k+1) = \mathbf{A}\mathbf{z}(k)$ is marginally stable or stable in the sense of Lyapunov if every finite state \mathbf{z}_0 excites a bounded response. It is asymptotically stable if every finite initial state excites a bounded response, which approaches $\mathbf{0}$ as $t \rightarrow \infty$.

Lema 3.2. System $\mathbf{z}(k+1) = \mathbf{A}\mathbf{z}(k)$ is marginally stable iff all eigenvalues of \mathbf{A} have magnitudes less than or equal to 1 and those equal to 1 are simple roots of the minimal polynomial of \mathbf{A} . It is asymptotically stable iff all eigenvalues of \mathbf{A} have magnitudes less than 1.

The eigenvalues of \mathbf{A} are as follows

$$\begin{cases} e_1 = \frac{\varphi_3 + 1 - \zeta + \sqrt{\Delta}}{2} \\ e_2 = \frac{\varphi_3 + 1 - \zeta - \sqrt{\Delta}}{2} \end{cases} \quad (21)$$

$$\Delta = (\varphi_3 + 1 - \zeta)^2 - 4\varphi_3 \quad (22)$$

When $\Delta < 0$, e_1 and e_2 is a pair of conjugate complex. $\|e_1\| = \|e_2\| = \sqrt{\varphi_3}$. If $0 \leq \sqrt{\varphi_3} < 1$, the system is convergent. When $\Delta \geq 0$, if both $\|e_1\|$ and $\|e_2\|$ are less than 1, the system is convergent. So, the convergence condition is the union of the two cases above, that is.

$$\mathcal{R}_z = \{\Delta < 0 \text{ and } \varphi_3 \in [0, 1)\} \cup \{\Delta \geq 0 \text{ and } \|e_{1,2}\| < 1\} \quad (23)$$

\mathcal{R}_z is termed the convergence region. $\|e_{1,2}\| < 1$ represents both $\|e_1\|$ and $\|e_2\|$ are less than 1.

To further examine the convergence, we can derive system Eq. (12) as follows

$$x(k+2) - (\varphi_3 - \zeta + 1)x(k+1) + \varphi_3x(k) = 0 \quad (24)$$

$$y(k) = \Gamma - \frac{x(k+1) - \varphi_3x(k)}{\zeta} \quad (25)$$

So, we obtain

$$x(k) = c_1e_1^k + c_2e_2^k \quad (26)$$

$$y(k) = \Gamma - \frac{c_1e_1^k(e_1 - \varphi_3) + c_2e_2^k(e_2 - \varphi_3)}{\zeta} \quad (27)$$

c_1 and c_2 are defined by initial state \mathbf{z}_0 . Since both $\|e_1\|$ and $\|e_2\|$ are less than 1 under the convergence condition, $\lim_{k \rightarrow \infty} x(k) = 0$, $\lim_{k \rightarrow \infty} y(k) = \Gamma$.

In order to investigate the convergence process, we set $\varphi_3 = 0$, $\zeta = 1.5$ to satisfy \mathcal{R}_z and test in the 2 dimensional Rastigin function. It's shown in Fig.2.

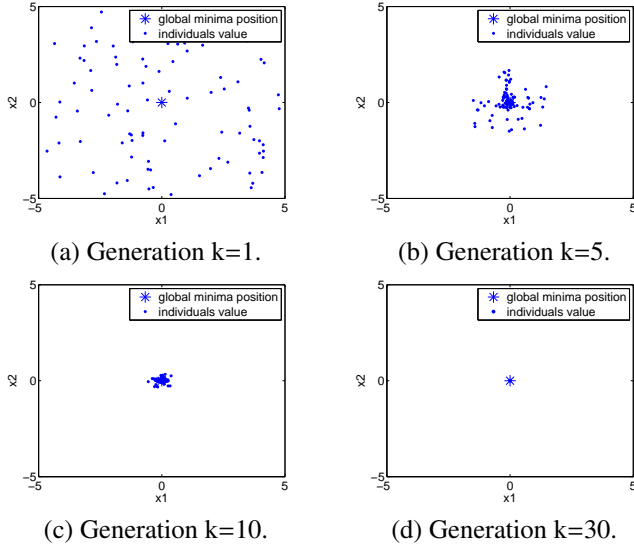


Figure 2: The convergence process of PDEA testing in the 2 dimensional Rastigrin function based on control theory.

3.3. Convergence condition relaxation

However, the convergence condition \mathcal{R}_z is too strong to be satisfied. Here we will relax it. Let's consider the main procedure of PDEA. At the beginning, the initial generation is created randomly. Each individual appears with the same probability p in the solution space. Then, the individual is driven by the strategy consisted of pd_c , pd_h , and pd_r to the convergence point. The closer the individual to the convergence point, the greater the probability that it appears in the area near the convergence point.

From the point of view of entropy theory, we define the entropy of the individual as $\log(1/p(i))$, where $p(i)$ denotes the probability of individual i appearing in a certain area of the solution space. $\log(1/p(i))$ represents the uncertainty of individual i . Therefore, the entropy of generation k can be defined as follows.

$$H(k) = E[\log(1/p_k(i))] = \sum_{i=1}^n p_k(i) \cdot \log(1/p_k(i)) \quad (28)$$

Then, the behaviour of the individual is analyzed by $H(k)$.

From the convergence condition, Eqs. (26) and (27) derived of system of Eq. (12), it shows that all the individuals converge to the point determined by Γ . To inspect the system, during the iterations, $x(k)$ trends to 0 and $y(k)$ trends to Γ . That reflects the variation tendency of individuals is that their positions are getting closer to Γ . The area where individuals appear is becoming more and more certain. Therefore, $H(k)$ is decreasing to 0, when $k \rightarrow \infty$.

However, the convergence condition that all the eigenvalues of \mathbf{A} have magnitudes less than 1 is too strong. Consider the general situation that $\mathbf{u} = (u_1, u_2, u_3)$ is some probability distributions. According to the analysis based on systems theory, PDEA is convergent if and only if all the range of the distribution falls fully within \mathcal{R}_z . However, in terms of the analysis

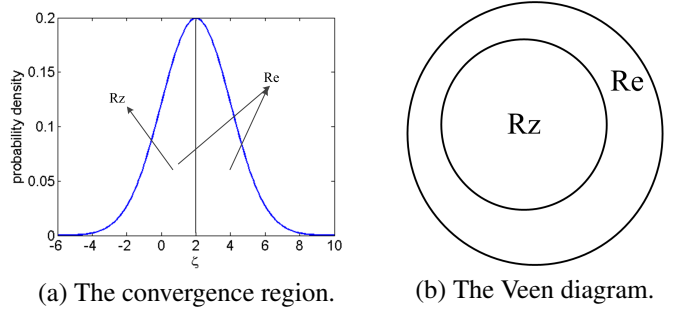


Figure 3: An example of illustration of the convergence region before and after relaxation. \mathcal{R}_z is the convergence region based on systems theory. \mathcal{R}_e is the convergence region based on entropy theory.

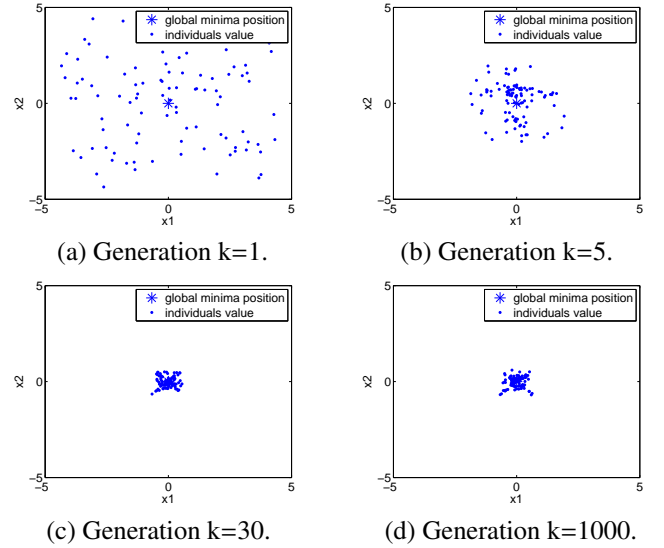


Figure 4: The convergence process of PDEA testing in the 2 dimensional Rastigrin function based on entropy theory.

based on entropy theory, the convergence condition is relaxed. Let's illustrate it by an example.

Consider the system of Eq. (12), the solution is

$$\begin{bmatrix} x(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} c_1 \\ -c_1(e_1 - \varphi_3) \\ \zeta \end{bmatrix} \begin{bmatrix} e_1^k \\ 1 \end{bmatrix} + \begin{bmatrix} c_2 \\ -c_2(e_2 - \varphi_3) \\ \zeta \end{bmatrix} \begin{bmatrix} e_2^k \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ \Gamma \end{bmatrix} \quad (29)$$

To inspect the convergence condition relaxation, we assume $\varphi_3 = 0$ and $\zeta \sim \mathcal{N}(2, 2)$, where $\mathcal{N}(2, 2)$ denotes the Gaussian distribution with the means and variance both equal to 2. Therefore, when $\zeta \geq 2$, $\|e_1\| = 0$ and $\|e_2\| \geq 1$. When $\zeta < 2$, $\|e_1\| < 1$ and $\|e_2\| < 1$. So, according to systems theory, if $\zeta < 2$, the system is convergent. If $\zeta \geq 2$, the system is not convergent. However, from the point of view of entropy theory, when $\zeta \geq 2$, $H(k)$ does not decrease. When $\zeta < 2$, $H(k)$ decreases. Because $E(\zeta < 2) = E(\zeta \geq 2) = 0.5$. So, $\lim_{k \rightarrow \infty} H(k) = 0.5H(1)$. The convergence region is illustrated in Fig. 3.

The convergence process based on entropy theory is plotted in Fig. 4. It is clear that even though individuals do not converge to a certain point, they converge to a certain area. Over the convergence process, $H(k)$ decreases to a certain value.

Using entropy theory, we derive the relaxed convergence

condition. If $H(k) < H(1)$ as $k \rightarrow \infty$, the system is convergent. This means that PDEA is convergent, iff $\mathcal{R}_Z \cap \mathcal{R}_e \neq \emptyset$.

However, in practice, to guarantee the performance of PDEA, we suggest keeping $\lim_{k \rightarrow \infty} H(k) \leq 0.5H(1)$.

4. Comparison with other EAs

Convergence analysis of EAs is an important and challenging issue. Markov chain model is proposed to study the convergence of GA. However, the convergence analysis is based on the assumption of infinite population and generation. PSO, BFO, and DE are analysed based on dynamic equations. However, the convergence condition of PSO is too strong, and the standard PSO is not convergent. The convergence analyses of BFO and DE are both based on the assumption that the objective function is differentiable.

Unlike the above analysis, PDEA convergence analysis does not require such strong assumptions. First, from the point of view of systems theory, we consider PDEA as a linear discrete-time state space system and prove that it is controllable. For PDEA, controllability is convergence. We derive the convergence condition of PDEA according to systems theory. Then we relax the convergence condition from the point of view of entropy theory. That can also be used to relax the convergence condition of DE, PSO, and BFO.

5. Numeric Evaluations

In this section, PDEA is applied to solve 16 benchmark functions [31] for numeric evaluations. These functions are selected from commonly used optimization test functions. Furthermore, PDEA is applied to an adaptive liquid level control system of a surge tank.

5.1. Evaluations with benchmark functions

The properties of these test functions are summarized in Table 1, where $f(x^*)$ denotes the optimum of the benchmark functions.

GA, as a popular EA, is used as the comparison reference for PDEA. The crossover, mutation probabilities and encode accuracy of GA are set as 0.9, 0.1, and 10^{-4} , respectively. The parameters of PDEA are set as below. α and β are determined by pdc and pdh . We set $\alpha/\beta = pdc/pdh$, $\alpha + \beta = 4$, $\max(\alpha/\beta) = 3$, $\min(\alpha/\beta) = 1/3$, and $\gamma = 1$. $\mathbf{u} = (u_1, u_2, u_3)$ obeys uniform distribution in $(0, 1)$.

The simulation environment is Matlab version 2010b, run in Operating System of MS Windows 10, 64-bit, Processor of Intel Core I7-4790, Memory of 8GB DDR3 RAM.

To make the tests complete, 50 independent tests are carried out for each benchmark function. The population size is set as 100, maximum iteration is set 200. The performance metrics, including computation time, mean error, and variance, are listed in Table 1.

From the numeric simulation results, it is clear that PDEA is capable of finding the optimal or near optimal solutions despite

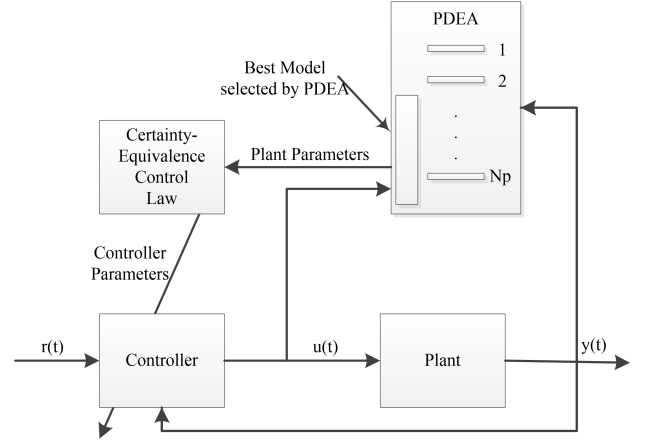


Figure 5: PDEA for adaptive control ($r(t)$ is the reference input)

of the diversity of benchmark functions, with GA as the comparison reference. In terms of the mean value, variance value, and computation time, the accuracy, reliability and computation speed of PDEA, are acceptable.

5.2. Adaptive liquid level control

Optimization methods can be used to estimate models and design controllers in adaptive control. Therefore, in order to evaluate the engineering performance, PDEA is applied to an adaptive liquid level control of a surge tank.

The purpose of applying PDEA in Fig. 5 is to learn the plant model during the operation of the indirect adaptive control system. To evaluate the learning capability of PDEA, the error between the model output and the plant output is the objective function as defined. During each interval, PDEA searches in the objective function space to find the plant model of minimum identification error. Then, the best model is selected and applied to the standard certainty-equivalence control law.

We consider the surge tank liquid level control problem as in [13]. The model is below

$$\frac{dh(t)}{dt} = \frac{-\bar{d}\sqrt{2gh(t)}}{A(h(t))} + \frac{\bar{c}}{A(h(t))}u(t) \quad (30)$$

where $h(t)$ is the liquid level, $u(t)$ is the input. $A(h(t)) = \|\bar{a}h(t) + \bar{b}\|$ represents the tank cross-sectional area; \bar{a} , \bar{b} , \bar{c} , and \bar{d} are constants. However, the tank cross-sectional area is not known. We have to estimate the plant dynamics so that control can be designed for compensation.

The parameters of PDEA are set as follows. $\mathbf{u} = (u_1, u_2, u_3)$ is uniform distribution in $(0, 1)$. $\alpha = \beta = 1$, and $\gamma = 0.5$, which means we set pdc and pdh with the same weight, pdr has a half weight. Population size $Np = 10$. The maximum number of iteration $Nmax = 10$.

Each individual corresponds to the model parameter. The objective function is defined as the sum of squares of the 50 past identification error values. For parameter adjustment, PDEA searches the objective function space to find the best plant parameters. The tracking performance and the best objective function value are plotted in Fig.6. The reference input of liquid

Table 1: PDEA performances on benchmark functions

Benchmark Function	Property	$f(x^*)$	GA			PDEA		
			Mean	Variance	Time (s)	Mean	Variance	Time (s)
Ackley	Many Local Minima	0	0.8167	1.4899	0.3860	0.0086	2.3880E-5	0.1850
Griewank		0	0.0700	0.0060	0.3470	0.0014	4.4960E-6	0.1980
Rastrigin		0	0.3419	0.2230	0.2910	0.0057	3.2567E-5	0.1590
Drop-Wave		-1	-0.9372	0.0020	0.3310	-0.9996	1.4190E-7	0.1810
Bohachevsky	Bowl Shaped	0	0.8386	3.0724	0.3260	8.9568E-4	9.1872E-7	0.1830
Sphere		0	8.6769E-4	1.4583E-5	0.3060	1.0455E-5	8.2029E-11	0.1550
Zakharow	Plate Shaped	0	0.0205	0.0048	0.3110	1.0539E-5	9.1300E-11	0.1890
Power Sum		0	0.1032	0.0189	0.6790	0.0171	1.7254E-4	0.4480
Easom	Steep Drops	-1	-0.0730	0.0012	0.3340	-0.9584	0.0391	0.170
Michalewicz		-1.8013	-1.7993	3.6263E-5	0.3100	-1.8011	2.4055E-8	0.2000
Six Hump	Valley Shaped	-1.0316	-1.0300	1.8835E-5	0.3190	-1.0316	1.4779E-9	0.1710
Dixon & Price		0	0.0292	0.0041	0.3210	3.7151E-5	1.5795E-9	0.1700
Rosenbrock's		0	0.2101	0.1591	0.3240	1.1036E-4	1.4905E-8	0.1710
Beale	Others	0	0.0127	0.0120	0.3130	1.6613E-5	2.0765E-10	0.1630
Branin		0.3978	0.4081	4.5087E-4	0.3080	0.3979	6.0497E-10	0.1690
Goldstein&Price		3	3.0962	0.0460	0.2950	3.0022	8.1864E-6	0.1790

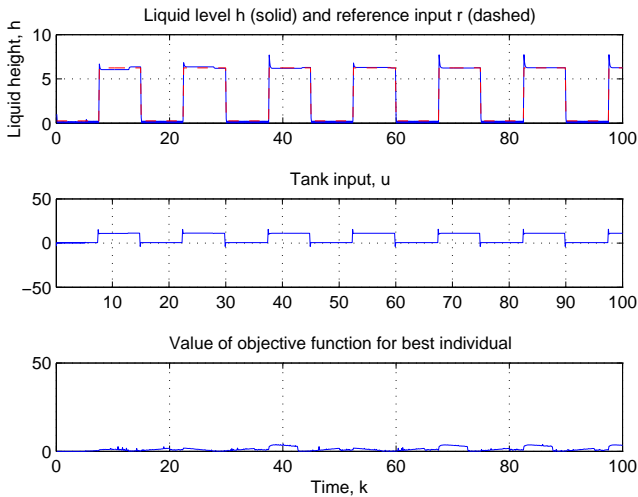


Figure 6: The response of adaptive controller.

level is a periodic square wave. For each square input, only at the first sampling interval, the tracking error is a little big. The maximum error value is 23.15%. But after 2 or 3 intervals, the tracking liquid level approaches the reference input quickly. The tracking error is almost zero.

6. Conclusions

In this paper, we have proposed a novel evolutionary algorithm called pseudo derivative evolutionary algorithm (PDEA). PDEA combines derivative with evolutionary algorithms. The advantage of PDEA is utilizing pseudo derivative to drive the optimization process. The convergence of PDEA is first analysed based on systems theory and the convergence condition is

derived. Next, this condition, which is too strong, is relaxed based on entropy theory. Different from common convergence analysis with other EAs, the convergence of PDEA does not require strong assumptions, such as differentiability and infinite population or generation. And the strict convergence region is also extended. Finally, we apply PDEA to both benchmark functions and adaptive control problem for evaluating the numeric and engineering performances. For the benchmark functions, PDEA shows strong ability in finding the optimal solution of different kinds of benchmark functions. For the adaptive control problem, PDEA is capable of identifying the plant model and tracking the reference input well.

References

- [1] P. Faliszewski, J. Sawicki, R. Schaefer, M. Smolka, Multiwinner voting in genetic algorithms, *IEEE Intelligent Systems* 32 (2017) 40–48.
- [2] A. L. Filho, D. Vettorazzi, A. Cruz, C. Lima, Atm: a new heuristic algorithm based on genetic algorithm and betting theory, *IEEE Latin America Transactions* 15 (2017) 510–516.
- [3] J. A. Martins, A. Mazayev, N. Correia, G. Schtz, A. Barradas, Gacn: Self-clustering genetic algorithm for constrained networks, *IEEE Communications Letters* 21 (2017) 628–631.
- [4] U. Bhowan, M. Johnston, M. Zhang, X. Yao, Evolving diverse ensembles using genetic programming for classification with unbalanced data, *IEEE Transactions on Evolutionary Computation* 17 (2013) 368–386.
- [5] U. Bhowan, M. Johnston, M. Zhang, X. Yao, Reusing genetic programming for ensemble selection in classification of unbalanced data, *IEEE Transactions on Evolutionary Computation* 18 (2014) 893–908.
- [6] C. L. P. Chen, T. Zhang, L. Chen, S. C. Tam, I-ching divination evolutionary algorithm and its convergence analysis, *IEEE Transactions on Cybernetics* 47 (2017) 2–13.
- [7] Y. L. Li, Z. H. Zhan, Y. J. Gong, J. Zhang, Y. Li, Q. Li, Fast micro-differential evolution for topological active net optimization., *IEEE transactions on cybernetics* 46 (2015) 1411.
- [8] Y. Del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, Particle swarm optimization: Basic concepts, variants and applications in power systems, *Evolutionary Computation IEEE Transactions on* 12 (2008) 171–195.

- [9] W. Hu, G. G. Yen, Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system, *IEEE Transactions on Evolutionary Computation* 19 (2015) 1–18.
- [10] D. H. Tungadio, B. P. Numbi, M. W. Siti, A. A. Jimoh, D. H. Tungadio, B. P. Numbi, M. W. Siti, Particle swarm optimization for power system state estimation, *Neurocomputing* 148 (2015) 175–180.
- [11] S. Dasgupta, S. Das, A. Abraham, A. Biswas, Adaptive computational chemotaxis in bacterial foraging optimization: An analysis, *IEEE Transactions on Evolutionary Computation* 13 (2009) 919–941.
- [12] J. Yi, D. Huang, S. Fu, H. He, T. Li, Multi-objective bacterial foraging optimization algorithm based on parallel cell entropy for aluminum electrolysis production process, *IEEE Transactions on Industrial Electronics* 63 (2016) 2488–2500.
- [13] K. M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, *IEEE Control Systems* 22 (2002) 52–67.
- [14] L. Xu, J. Yeming, H. Ming, Solving hybrid flow-shop scheduling based on improved multi-objective artificial bee colony algorithm, in: 2016 2nd International Conference on Cloud Computing and Internet of Things (CCIoT), 2016, pp. 43–47. doi:10.1109/CCIoT.2016.7868300.
- [15] Y. Liu, C. Gao, Z. Zhang, Y. Lu, S. Chen, M. Liang, L. Tao, Solving np-hard problems with physarum-based ant colony system, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 14 (2017) 108–120.
- [16] A. M. Shaheen, R. A. El-Sehiemy, S. M. Farrag, Solving multi-objective optimal power flow problem via forced initialised differential evolution algorithm, *IET Generation, Transmission Distribution* 10 (2016) 1634–1647.
- [17] H. Li, L. Zhang, Y. Jiao, Discrete differential evolution algorithm for integer linear bilevel programming problems, *Journal of Systems Engineering and Electronics* 27 (2016) 912–919.
- [18] V. Santucci, M. Baiocchi, A. Milani, Algebraic differential evolution algorithm for the permutation flowshop scheduling problem with total flowtime criterion, *IEEE Transactions on Evolutionary Computation* 20 (2016) 682–694.
- [19] D. E. Goldberg, P. Segrest, Finite markov chain analysis of genetic algorithms, in: *International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, 1987.
- [20] A. E. Eiben, E. H. L. Aarts, K. M. V. Hee, Global convergence of genetic algorithms: A markov chain analysis, Springer Berlin Heidelberg, 1990.
- [21] G. Rudolph, Convergence analysis of canonical genetic algorithms, *IEEE Transactions on Neural Networks* 5 (1994) 96–101.
- [22] X. Qi, F. Palmieri, Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space. part i: Basic properties of selection and mutation, *Neural Networks IEEE Transactions on* 5 (1994) 102–119.
- [23] R. C. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, in: *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, 2000, pp. 84–88 vol.1.
- [24] M. Clerc, J. Kennedy, The particle swarm - explosion, stability, and convergence in a multidimensional complex space, *Evolutionary Computation IEEE Transactions on* 6 (2002) 58–73.
- [25] I. C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, *Information Processing Letters* 85 (2003) 317–325.
- [26] F. Van, den Bergh, A. P. Engelbrecht, A new locally convergent particle swarm optimiser, in: *IEEE International Conference on Systems, Man and Cybernetics*, 2002, p. 6 pp. vol.3.
- [27] S. Das, S. Dasgupta, A. Biswas, A. Abraham, A. Konar, On stability of the chemotactic dynamics in bacterial-foraging optimization algorithm, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 39 (2009) 670–679.
- [28] S. Dasgupta, A. Biswas, S. Das, A. Abraham, The population dynamics of differential evolution: A mathematical model, in: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1439–1446. doi:10.1109/CEC.2008.4630983.
- [29] S. Dasgupta, S. Das, A. Biswas, A. Abraham, On stability and convergence of the population-dynamics in differential evolution, *Ai Communications* 22 (2009) 1–20.
- [30] A. K. Qin, V. L. Huang, P. N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, *IEEE Transactions on Evolutionary Computation* 13 (2009) 398–417.
- [31] D. B. S. Surjanovic, Virtual library of simulation experiments: Test functions and datasets, <https://www.sfu.ca/~ssurjano/optimization.html>, 2017. Accessed April 4, 2017.